# Halting Problem for One-State Turing Machines

Yannick Saouter

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Halting problem for one-state Turing machines*

Yannick Saouter

**N° 2577**

Juin 1995

PROGRAMME 1


ISSN 0249-6399

INRIA
RENNES

# Halting problem for one-state Turing machines

Yannick Saouter

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet API

Rapport de recherche n ° 2577 — Juin 1995 — 18 pages

**Abstract:** In [1], Turing has established the well-known result of the indecidability of the general halting problem of Turing machines. It is also well known that there exists entire classes of Turing machines whose halting is decidable (machines operating on an alphabet of one symbol, machines with finite tapes etc ...). In this article we are paying attention on Turing machines with a single state for which we prove that the halting is decidable. The interest of this problème is that it settles exactly with other works the limit between decidable and undecidable in regard of the number of internal states of the machine.

**Key-words:** Turing Machines; Decidability

*(Résumé : tsvp)*

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

# Problème de l'arrêt pour machines de Turing à un état

**Résumé :** Dans [1], Turing a établi le résultat bien connu de l'indécidabilité de l'arrêt en général des machines de Turing. Il est bien connu aussi qu'il existe des classes entières de machines de Turing dont l'arrêt soit décidable (machines opérant sur un alphabet à une lettre, machines à bande finie etc ...). Dans cet article nous portons notre attention sur les machines de Turing à un état pour lesquelles nous prouvons que l'arrêt est décidable. L'intérêt de ce problème est qu'il établit avec exactitude avec d'autres travaux la limite entre décidable et indécidable du point de vue du nombre d'états internes d'une machine.

**Mots-clé :** Machines de Turing; Décidabilité

# 1 Introduction

In [1], Turing has established the well-known result of the indecidability of the general halting problem of Turing machines. Some works [2, 3] have posed the problem of determining the boundaries between decidable and undecidable. Indeed, in these articles, the authors use the concept of colors for the Turing machines operating on the alphabet $\{0, 1\}$ and explorates the problem of determining the minimum number of states which are necessary to render the halting problem undecidable. The main results of these articles is the decidability for the two-colors Turing machines and for the restricted class of four-colors non-erasing Turing machine, and the undecidability for the general three-colors Turing machines.

In this article we explorate another limitation of the concept of Turing machines: the number of non final states. The main results of this paper are the decidability for the one-state Turing machines and the undecidability for the six-states Turing machines. Moreover we prove that the concept of one-state Turing machines is strictly orthogonal to the one of finite automata.

# 2 Definitions

Throughout this paper many notions will be addressed : the reader will be supposed to be especially familiar with the topics of strings, Turing machines and regular languages (see [4] for details). In this paragraph, we develop the notion of Turing language.

Classicaly[4], the Turing languages are defined the following way:

**Definition 2.1** (Turing language) *Let $M$ be a Turing machine whose alphabet is $A$ and whose configurations are denoted $(u, q, a, v)$. The* Turing language *recognized by $M$ is the set of the strings $a.v$ of $A^*$ for which $M$ stops when beginning in the initial configuration $(\varepsilon, q_0, a, v)$.*

This definition is unsufficient for one-state machine. Indeed, let us consider the example depicted in [5, p. 461] of a language $L = \{a^n.b^n.c^n \mid n \in \mathbf{N}\}$. The author describes a machine which has the following behavior: it replaces the first $a$ letter encountered by a new symbol $x$, then it skips all the other following $a$ symbols. When it encounters a $b$ it replaces it again by $x$ and then skip all the following $b$ to replace the first $c$ by $x$. At this moment the machine returns back to the first $a$ of the band and do a cycle again. The machine then stops if at the end there is only $x$ symbols on the tape, that is to say that if the tape contains at least one other symbol, the machine is designed in a way that it diverges. The main problem with this approach is that when introducing a new symbol (here $x$), one have to ensure that no string containing this new symbol is recognized by the machine.

In fact that is an error-recovery problem. After having skipped all the $a$ symbols, the machine may encounter either a $b$ or an $x$. In the first phase if the head encounters a symbol different of $b$ then an error have to be noticed and the machine might not stop. In the following phases, a symbol $x$ encountered should not give an error. So we have two different behaviors depending on the number of the phase. Typically this differenciation is done by assuming two different states: one for the first phase and one for the others. At this point it is clear that if we do not broaden the notion of Turing languages for one-state machines, very few languages will be recognized. So we define:

**Definition 2.2** (Turing language of one-state machine) *Let $L$ be a language of an alphabet $A$. It will be said a* Turing language of one-state machine *if there exists an alphabet $B$, such that $A \subset B$ and a one-state Turing machine $M$ of alphabet $B$, with configurations*

*$(u, q, a, v)$, such that the set of strings $a.v$ of $A^*$ for which $M$ stops when beginning in the initial configuration $(\varepsilon, q_0, a, v)$ is equal to $L$.*

This definition enables then to introduce special characters which will not be taken in account when treating of the recognized language. We will classically denote $\flat$ the separator of the Turing machines and $\varepsilon$ the null string. We will also denote $l(s)$ the length of a string $s$. The rewritings will be denoted as triples $(u, a, v)$ denoting respectively the string on the left, the symbol under the head and the string on the right of the head. The one-step (resp. $m$-step) transition relation will be denoted $\xrightarrow{1}$ (resp. $\xrightarrow{m}$), and the transitive closure of this relation will be $\xrightarrow{*}$.

# 3 Halting problem for one-state machines

The main result of this section is to establish that the halting problem for Turing machines with only one non-final states is decidable. But at first we are going to precise the set of languages recognized by those machines. We will establish that it includes a strict subset of the regular languages, but also that it contains some other languages that are not regular.

## 3.1 One-state machines and regular languages

This section is devoted to make a comparison between the regular languages and the one-state Turing languages. In theorem 3.1 we exhibit a (small) class of regular languages which are recognized with one-state Turing machines. Theorems 3.2 and 3.3 shows that there exists one-state Turing languages which are not regular and finally theorem 3.2 shows conversely that there exists regular languages which are not one-state Turing languages. Then we are faced with two distinct classes of languages whose intersection is not empty. The following theorems also illustrate the great difficulty of the characterization of one-state Turing languages.

**Fact 3.1** *Let $A$ be an alphabet and let $a \in A$ such that $A - a \neq \emptyset$. Then any language of the form $a^k.(a)^*$ with $k \geq 1$ is a one-state Turing language.*

**Proof** Let us pose $A = \{\flat, a, b_1, b_2, ..., b_l\}$. We will introduce the additionnal symbols $c_1, c_2, ..., c_k, d$ and $e$. Then we design a Turing machine $M$ whose alphabet is $\{\flat, a, b_1, b_2, ..., b_l, c_1, c_2, ..., c_k\}$. The transition function of the machine $M$ is the following one:

$$
\begin{aligned}
t(\flat) &= (c_1, G) && (1)\\
t(a) &= (d, D) && (2)\\
t(b_1) &= (b_1, G) && (3)\\
t(b_2) &= (b_2, G) && (4)\\
&\ldots && (5)\\
t(b_l) &= (b_l, G) && (6)\\
t(c_1) &= (c_2, G) && (7)\\
t(c_2) &= (c_3, G) && (8)\\
&\ldots && (9)\\
t(c_{k-1}) &= (c_k, G) && (10)\\
t(c_k) &= \textbf{stop} && (11)
\end{aligned}
$$

$$t(d) = (e, D) \quad (12)$$
$$t(e) = (e, G) \quad (13)$$

In order to stop, in every case, the machine has at least to attain the first symbol $\flat$ at the right of the head. Indeed the machine does have only one halting symbol, namely $c_k$, which belongs to the successors of $\flat$ by the transition function and of no other symbol of $A$. Now if the machine attains the character $\flat$ on the left of the initial position of the head, the machine clearly diverges by the left. So if we suppose the machine to stop, necessarily the machine has previously attained the first $\flat$ symbol on the right of the head. If the tape contain at least one cell with character $b_i$, then the head cannot attain a position at the right of this cell, according to the transition function. As a consequence the machine cannot stop. As well if the tape is empty, since $\flat$ is not a terminating character, then the machine diverges. So we have already proven that the language recognized by this machine is included in $(a)^+$.

So let $(\varepsilon, a, a^m)$ with $m \geq 0$ be the initial configuration of $M$. After $m+1$ rewritings we are led to the configuration $(d^{m+1}, \flat, \varepsilon)$. Then after two more rewritings we are led to $(d^m.e, c_1, \varepsilon)$. If $k = 1$ then $c_1$ is a terminating character and so the machine stops. We have then the result that $M$ recognizes $(a)^+$. If we have $k > 1$, an easy recurrence shows that the machine assumes after a finite number of rewriting each of the configurations $(d^{m-l+1}.e^l, c_l, \varepsilon)$, where $l \leq m$ and $l \leq k$.

If we have $m+1 < k$ then the machine assumes at one step the configuration $(e^{m+1}, c_{m+1}, \varepsilon)$. In this case $c_{m+1}$ is not a terminating symbol and the reader may easily figure that the machine runs out to infinity by the left.

If we have $m + 1 \geq k$ then the machine assumes the configuration $(d^{m+1-k}.e^k, c_k, \varepsilon)$ at one step and then stops at the following rewriting.

So the machine exactly recognizes the strings $a.a^m$ where $m + 1 \geq k$, i.e. the machine recognizes exactly $a^k.(a^*)$. □

The rest of this section will be devoted to prove that this result is not optimal. This will be in two steps. First of all, we have:

**Fact 3.2** *Let $A = \{a, b\}$ be an alphabet. We define the language $L$ as the set of the strings $m_1.m_2.\ \ldots\ .m_l$ such that:*

- *for all $i$, $m_i$ is of the form $a^{n_i}.b^{m_i}$;*
- *for all $n$, $1 \leq p \leq l$, $\Sigma_{i=1}^{p} n_i > \Sigma_{i=1}^{p} m_i$.*

*Then the language $L$ is recognized by a one-state Turing machine.*

**Proof** The alphabet of the Turing machine recognizing this language will be $B = \{\flat, a, b, c, d, e, f, g\}$. The function of transition $t$ will be defined as follows:

$$t(\flat) = (g, L) \quad (14)$$
$$t(a) = (c, R) \quad (15)$$
$$t(b) = (d, L) \quad (16)$$
$$t(c) = (e, R) \quad (17)$$
$$t(d) = (e, R) \quad (18)$$
$$t(e) = (f, L) \quad (19)$$
$$t(f) = (e, R) \quad (20)$$
$$t(g) = \textbf{stop} \quad (21)$$

In a sake of simplicity the configurations of the form $(u, a, v)$ will be denoted here $[u, a.v]$. The reader may easily prove the following facts:

$$\forall k \geq 0,\ [e^k, e.v'] \quad \textbf{diverges} \tag{22}$$

$$[\varepsilon, \flat, v'] \quad \textbf{diverges} \tag{23}$$

$$\forall n \geq 0,\ [v, a^n.v'] \xrightarrow{n} [v.c^n, v'] \tag{24}$$

$$\forall k \geq 0,\ m \geq 0,\ [v.c^j.e^k, b^m.v'] \xrightarrow{*} [v.c^{j-m}.e^{k+2m}, v'] \textbf{ if } j \geq m \tag{25}$$

$$\xrightarrow{*} [v.e^k + 2j, b^{m-j}.v'] \tag{26}$$

$$[v.c.e^k, \varepsilon] \quad \textbf{halts} \tag{27}$$

Let $u$ be a string recognized by this machine. We will pose $u = a^n.b^m.w$ where $w$ does not have $b$ as prefix and is equal to $\varepsilon$ if $m = 0$. We study the rewritings of the tape from a configuration of the form $[v, a^n.b^m.w]$ where $v$ is a string of the form $v = c^{j_1}.e^{k_1}.c^{j_2}.e^{k_2}. \ldots .c^{j_N}.e^{k_N}$ where some of the indices $k_i$ and $j_i$ are eventually null.

According the rules (24-26) we have then:

$$\begin{aligned} [v, a^n.b^m.v'] &\xrightarrow{*} [v.c^{n-m}.e^{2m}, v'] \textbf{ if } n \geq m \\ &\xrightarrow{*} [v.e^{2n}, b^{m-n}.v'] \textbf{ if } n \leq m \end{aligned}$$

At this point, if $n \leq m$, we have by recurrence over the $j_i$ with the rules (25-26):

$$\begin{aligned} [v''.c^{j_N}.e^{2n+k_N}, b^{m-n}.v'] &\xrightarrow{*} [v''.c^{j_N+n-m}.e^{2m+k_N}, w] \textbf{ if } j_N \geq m-n \\ &\xrightarrow{*} [v''.e^{2n+k_N+2j_N}, b^{m-n-j_N}.w] \textbf{ if } j_N \leq m-n \end{aligned}$$

and:

$$\begin{aligned} [v, a^n.b^m.v'] &\xrightarrow{*} [e^{2n+\Sigma_{i=1}^N k_i + 2\Sigma_{i=1}^N j_i}, b^{m-n-\Sigma_{i=1}^N j_i}.v'] \textbf{ if } m-n \geq \Sigma_{i=1}^N j_i \\ &\xrightarrow{*} [v'''.c^{\Sigma_{i=I}^N j_i + n - m}.e^{2m+\Sigma_{i=I}^N k_i}, v'] \end{aligned}$$

where $I$ is the maximal index such that $\Sigma_{i=I}^N j_i > m - n$.

So according to rule 22, the machines diverges if $m - n \geq \Sigma_{i=1}^N j_i$ and the configuration rewrites into configuration $[c^{j_1}.e^{k_1}.\ldots.c^{j_{I-1}}.e^{k_{I-1}}.c^{\Sigma_{i=I}^N j_i - n + m}.e^{2m+\Sigma_{i=I}^N k_i}, v']$ otherwise, which is an halting configuration according rule (27).

Let us pose now $u = a^{n_1}.b^{m_1}.\ldots.a^{n_l}.b^{m_l}$ then by recurrence, with the previous result, it is easy to show that the machine diverges as soon as there exists an index $L$ such that $\Sigma_{i=1}^L (m_i - n_i) \geq \Sigma_{i=1}^N j_i$ and halts otherwise.

At the beginning, we have $j_i = 0$ for all $i$, the recognized strings are exactly the strings $a^{n_1}.b^{m_1}.\ldots.a^{n_l}.b^{m_l}$ such that $\Sigma_{i=1}^L m_i < \Sigma_{i=1}^L n_i$ for all $1 \leq L \leq l$. □

This theorem, together with the following one will prove that the concept of one-state Turing machine is not equivalent to finite automata, and so in particular that the halting problem for these machines does not reduce to the one of automata. Indeed, we have:

**Fact 3.3** *The language $L$ of theorem 3.2 is not regular.*

**Proof** Classical proof. □

The theorem 3.1 proves that a subset of regular languages are one-state Turing languages. Morevover theorems 3.2 and 3.3 prove that some of the one-state Turing languages are not regular. One natural question is then to know whether any regular language is a one-state Turing language. The answer is negative :

**Theorem 3.1** *Let $A = \{a, b\}$ be an alphabet. Then the language $(a)^*$ is not a one-state Turing language.*

**Proof** This theorem is easy: a machine that recognizes $(a)^*$, must recognize $\varepsilon$. That implies that $\flat$ is a halting symbol. But since $b$ is not recognized, it is not an halting symbol. At this point whatever is the transition associated with $b$, the machine recognizes also the string $b^1$. □

We see then that for a one-state Turing machine, the fact of recognizing the string $\varepsilon$ is source of problems. The following theorem proves that relaxing this constraint is not sufficient :

**Theorem 3.2** *Let $A = \{a, b\}$ be an alphabet. Then the language $(a)^+ \cup (b)^+ \cup b.(a)^+$ is not a one-state Turing language.*

**Proof** Let $M$ be the machine that would recognize this language. We have:

- $a$ is not an halting symbol (otherwise $aA^*$ would be entirely recognized).
- $b$ is not an halting symbol.
- $\flat$ is not an halting symbol (otherwise $\varepsilon$ would be recognized).
- the associated movement for $a$ is not $L$. Indeed in this case the movement associated to $\flat$ is necessarily $R$ in order that the machine recognize $a$. But at this point since the machine is supposed to recognized $a$, the reader may easily prove that the machine would recognize any string beginning with $a$.
- the associated movement for $b$ is not $L$.

So necessarily, $t(\flat) = (c, L)$, $t(a) = (d, R)$ and $t(b) = (e, R)$.

When $M$ recognizes a string of $(a)^+$, with our informations, we see that $M$ skips by the right the symbols $a$ it encounters, replacing them by symbols $d$, until it reaches $\flat$, which is replaced by a symbol $c$ and makes one left movement. So after a finite number of steps the tape is something like illustrated in figure 1.

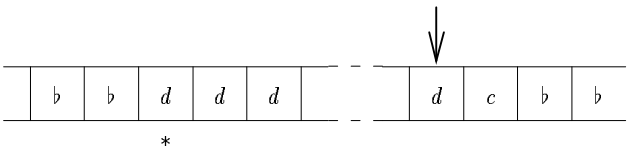


Figure 1: An intermediate configuration

In the following configurations, the leftmost position of the head of the machine will be the cell marked by the star. Indeed, if the head reaches a cell at the left of the starred cell then the machine runs to infinity by the left since $t(\flat) = (c, L)$. But we know that $M$ stops, so we may affirm that $M$ in a configuration as illustrated in figure 1 stops and its head never attain a position on the left of the starred cell.

But now, it is easy to prove that such a machine would also recognize any string of $(b)^+.(a)^+$. Indeed if $M$ is given such a string, by simulation, after a finite number of steps

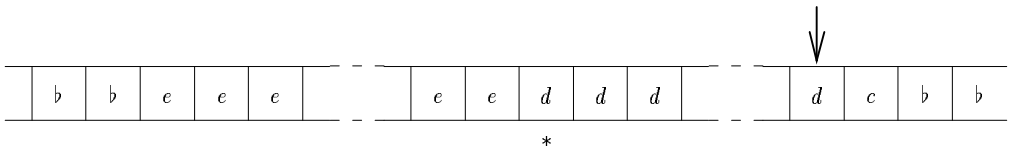


Figure 2: Configuration for $(b)^+.(a)^+$

the machine has a global configuration as illustrated in figure 2. From what precedes, the head cannot reach a position at the left of the starred cell and so the global behavior of this configuration is identical to the one of figure 1, and so the machine stops. Then a one-state Turing machine which recognize $(a)^+$ and $(b)^+$, also recognizes $(b)^+.(a)^+$ (and also $(a)^+.(b)^+$ by symetry) Then the regular language $(a)^+ \cup (b)^+ \cup b.(a)^+$ is not a one-state Turing language. □

## 3.2 Halting problem of one state Turing machines

This section is devoted to the main result of this article:

**Theorem 3.3** *The halting problem for the one-state Turing machines is decidable.*

This theorem will be proven in three steps, according the properties of $\flat$. We will distinguish the cases where respectively, $\flat$ is an halting symbol for the machine, the orbit, in terms of the function of transition (see below for definition), of $\flat$ contains an halting symbol and the general case.

As we have seen, we need a new definition:

**Definition 3.1** (orbit) *Let $M$ be a one-state Turing machine of alphabet $A$ and of transition function $t$. We pose for any non halting symbol $x$ in $A$, $t(x) = (c(x), m(x))$ where $c(x)$ is the character written on the tape by the head and $m(x)$ is the movement performed by the head after having read the symbol $m(x)$ from the tape. Let $a$ be a symbol of $A$. Then the* orbit *of $a$ is the series $a, c(a), c^2(a), ..., c^n(a), ....$ The orbit is finite if and only if $c^n(a)$ is an halting symbol for some $n$ and is periodic (but not necessarily cyclic) on the case of the contrary.*

At the beginning of the simulation we have then a configuration of the form $(\varepsilon, b, u)$ where $a \in A$ and $u \in A^*$.

**Case when $\flat$ is an halting symbol** This case is very easy. Indeed as soon as the head reaches a cell not corresponding to a character of the string $b.u$, the machine halts. As a consequence, if the machine does not stop, necessarily it loops. But deciding whether a Turing machine loops (resp. stops) is known to be semi-decidable. Since these two cases, here converse of each other, are semi-decidable, they are both decidable.

[1] The reader may remark that we can overcome this problem by introducing the concept of rejecting symbol but this notion can be simulated by adding two new non-final states to the machine and so is out of the scope of this study.

**Case when $\flat$ is not an halting symbol** In this case, one more behavior of global comportment of the machine is possible. Indeed, on the contrary of the preceding case, the machine may diverge. Let us suppose that $t(\flat) = (a, L)$. In this case, whenever the machine attains a configuration of the form $(\varepsilon, \flat, s)$, it diverges by the left. Conversely if $t(\flat) = (a, R)$ then the machine diverges by the right whenever it attains a configuration of the type $(s, \flat, \varepsilon)$. In the case where $\flat$ is not an halting symbol, the study splits into the two following cases. Indeed, either the orbit of $\flat$ is finite (i.e. it contains an halting symbol), either this it contains a cycle.

The halting problem is not trivial. Indeed, many naive criteria fail. For instance, even if the orbit of $\flat$ is finite, the machine may diverge.

In the following we will always get our transition function such that $t(\flat) = (a, L)$. Of course the results we present are also valid for Turing machines for which $t(\flat) = (a, R)$: it just suffies to transpose both directions in the criteria.

The cells of our machine will be divided in three sets: the dead zone, the initial zone and the divergence zone.

**Definition 3.2** dead — initial — divergence zone *Let $M$ be a one-state Turing machine such that $t(\flat) = (a, L)$. Let $(\varepsilon, b, u)$ be its initial configuration. Then the cells corresponding to the string $b.u$ will form the* initial zone. *The set of the cells at the left (resp. right) of the initial zone, will be called* dead zone *(resp.* divergence zone*).*

It is clear that since we have $t(\flat) = (a, L)$, as soon as the head of the machine enters the dead zone, then the machine diverges by the left. If the machine enters the divergence zone it may diverge by the right or enter, eventually an infinite number of times, by the right the initial zone. In fact it is easy to build examples in which machines have both behaviors.

We will study the infinite behavior of the divergence zone: we will suppose that during the rewritings, the head of the machine is in the divergence zone an infinite number of times and we will show that it is decidable to know whether the machine reenters the initial zone after this position.

A general theorem will be of great help for this problem:

**Theorem 3.4** *Let $M$ be a one-state Turing machine of alphabet $A$ whose initial configuration is $(\varepsilon, b, u)$ where $b$ is a symbol of $A$ and $u$ a string of $A^*$. Let us suppose that the machine attains a configuration of the form $(u'.s, \flat, \varepsilon)$ where $u'$ is a string of $A^*$ such that $l(u') = 1 + l(u)$. Then the string $s$ does not depend on the string $b.u$, but only of the transition table of $M$.*

**Proof** This theorem can be proven by recurrence on the length $l$ of the string $s$.

If $l = 1$ the configuration attained by the machine is of the form $(u'.c, \flat, \varepsilon)$ where $c$ is a symbol of $A$. Then clearly $c$ is the first symbol in the orbit of $\flat$ such that $m(c) = R$. As we see the symbol $c$ does not depend on the string $b.u$.

Let us suppose that the property is true for length $l$. At the rank $l + 1$, we have then $(\varepsilon, b, u) \xrightarrow{*} (u_1.s_1, \flat, \varepsilon) \xrightarrow{*} (u_2.c_2.s_2, \flat, \varepsilon)$ with the strings $s_1$ and $s_2$ of length equal to $l$. Since the property is true at the rank $l$, we can say that the string $s_2$ does not depend on the initial string whose cells correspond to $u_2.c_2$ i.e. $b.u.\flat$, and so $s_2$ does not depend on $b.u$. By using the argument of recurrence, we have also that the string $s_1$ does not depend on the string $b.u$. Let us suppose that between the two last configurations the machine enters $k$ time the initial zone then $c_2$ is the $k$-th successor of $c_1$ in the orbit of $\flat$, whose associated movement is $L$, where $c_1$ is the first symbol of string $s_1$. Clearly the number of times the head enters the initial zone only depends on the string $s_1$ and so $c_2$ only depends on $s_1$. This

latter fact establishes the theorem. □

**Case when the orbit of $\flat$ contains an halting symbol** The previous theorem suffice in itself to make the decision of this case.

**Theorem 3.5** *Let $M$ be a Turing machine, such that the orbit of its separator contains an halting symbol. Then it is algorithmically possible to decide of the halting of $M$ from any initial configuration.*

**Proof** If after a finite number of transitions the machine attains the dead zone, as we have seen, it diverges by the left. If we suppose that the machine assumes infinitely many configuration with the head in the divergence zone, then it cannot assume also an infinite number of configurations with the head inside the initial zone. Indeed, at every time that the machine leaves the divergence zone to enter the initial zone, the leftmost symbol of the divergence zone is replaced by its successor in his orbit. But this symbol necessarily belongs to the orbit of $\flat$. So since the orbit of $\flat$ contains an halting symbol a machine with such a behavior necessarily halts which is in contradiction with the hypothesis.

So we have established the following alternatives of behaviors: the machine $M$ either stops, or attain an infinite number of configurations with the head in only one of the three zones of the tapes.

For the initial zone, the criterion is simple. Indeed from what precedes if the machine attains an infinite number of configurations in the initial zone, then after a finite number of rewritings the head leaves no more this zone. But this zone is finite, so in this case, the machine either stops or loops and both cases are decidable as previously.

For the divergence zone, a precise criterion clear out the problem. Let be $k$ the length of the orbit of $\flat$ (i.e. the number of symbols in this orbit). The machine cannot leave the divergence zone more than $k$ times if it does not stop.

So after a finite number of rewritings, from an initial configuration $(\varepsilon, b, u)$ the machine attains a configuration of the form $(w_1.s_1, \flat, \varepsilon)$, where the string $w_1$ corresponds to the initial zone, and after the head do not leave yet the divergence zone. In the first subcase the head of the machine stays on the cells corresponding to the string $s_1$. Then as previously, in this case, the machine either stops or loops and both cases are decidable.

In the second subcase, the machine attains a configuration of the form $(w_2.s_2, \flat, \varepsilon)$ with $l(s_2) = l(s_1)+1$. At this point theorem 3.4 applies, and we can certify that the strings $s_1$ and $s_2$ does not depend on the string $b.u$. Let us pose $s_2 = c.s$. Then by theorem 3.4, $s$ does not depend on the string $b.u.\flat$ but only of the transition function of $M$. As a consequence, since $l(s) = l(s_1)$, we have $s = s_1$. But, if the divergence zone ends by $s_1$, the machine cannot reaches a cell at the left of $s_1$, by hypothesis, and so a trivial recurrence allows to deduce that the machine diverges by the right and attains all the configurations of the form $(w.b^n.s_1, \flat, \varepsilon)$ where $n$ is any integer. Reciprocally if from a configuration of the type $(w.s_1, \flat, \varepsilon)$ it attains after a finite number of rewritings, a configuration of the type $(w.b.s_1, \flat, \varepsilon)$ without leaving the divergence zone then the machine diverges by the right. This latter result establishes the theorem. □

Let us summarize the criteria of the latter theorem:

- the machine may stop;
- the machine may loop;

- the head of the machine may attain the dead zone. In such a case, the machine diverges by the left;
- the machine attains two configurations $(w_1.s_1, \flat, \varepsilon)$ and $(w_2.c.s_1, \flat, \varepsilon)$ where $w_1$ and $w_2$ correspond to the initial zone and such that during all the rewritings between these two configurations the head does not attain the the initial zone. In this case the machine diverges by the right.

This theorem gives in fact an explicit criterion: indeed all the four preceding cases are semi-decidable and whence here decidable. Informally said, it suffices to simulate the machine until one of these conditions is met. We remark also that we have a precise description of the behavior of the machine.

**Case when the orbit of $\flat$ contains a cycle** This case will be dealt by using the same ideas but by examining more precisely the behavior of the machine. In fact there is one additionnal case that may happen: the head may enter an infinite number of times the initial zone and however diverges by the right. The purpose of this section is at first, to prove that the four previous cases and this latter behavior describe all the possible cases for this kind of Turing machines and secondly to give a criterion to detect latter this kind of behavior. As in the previous paragraph, we have the theorem:

**Theorem 3.6** *Let $M$ be a Turing machine, such that the orbit of its separator contains a cycle. Then it is algorithmically possible to decide of the halting of $M$ from any initial configuration.*

The proof is here a little bit longer than in the previous case. First of all, we will deal with two easy subcases:

**Proof of theorem 3.6** The case when the movements associated to every symbol of the cycle are equal to $L$ is easy to treat. Indeed in this case the fifth case of figure is impossible. If the head enters the initial zone and the divergence zone an infinite number of times, the leftmost symbol of the divergence zone after a finite number of steps belongs to the cycle. Then after the following entrance of the initial zone the head cannot attain anymore cells on the right of this cell, which contradicts the hypothesis. Then the halting of the machine can be decided by theorem 3.5.

A similar reasoning establishes that the fifth case is also impossible when the movements associated to every symbol of the cycle are equal to $R$: we are led to a same kind of contradiction by proving in this case that the head is stuck in the divergence zone after a finite number of steps and again decision can be made with only the theorem 3.5. □

So in the following, we will suppose that the cycle of the machine contains at least one symbol associated with each of the two movements. For the general case, we will establish at first some lemmas. We also need to introduce two functions on Turing machines:

**Definition 3.3** *Let $M$ be a one-state Turing machine such that the orbit of $\flat$ contains a cycle. We will note $R(M)$ (resp. $L(M)$) the number of symbols of the cycle of the orbit of $\flat$ whose associated movement is $R$ (resp. $L$).*

In our study of this case we will suppose that the machine assumes an infinite number of configurations in the divergence zone. Indeed, if the machine attains only a finite number of such configurations, then theorem 3.5 applies since nothing is supposed on the cycle of the machine for the three first cases of theorem 3.5.

We will need several definitions:

**Definition 3.4** *Let $O = \{o_i\}$ be the ordered graph of the orbit of $\flat$. We have $o_1 = \flat$, $o_2 = c(\flat)$, ..., $o_n = c^n(\flat)$ and $c(o_n) = o_{n-l+1}$, where $l$ is the number of symbols in the cycle of the orbit. We introduce the function next from $O$ into $O$ defined by:*

$$next(o_i) = min(\{k \mid k \geq 1, m \circ c^{k-1}(o_i) = R\})$$

*We also define the function succ from $O$ into $O$ by:*

$$\begin{aligned} succ(o_i) &= o_{m_i} \\ m_i &= min\{l \mid t \circ c^{l-1}(o_i) = (o_{m_i}, R)\} \end{aligned}$$

Informally said, $next(o_i)$ is the number of steps that have to be made on the orbit of $\flat$ when beginning at $o_i$ before to attain a symbol whose associated movement is $R$, while $succ(o_i)$ is the first symbol following $o_i$ in the orbit and such that its associated movement is $R$. We note also that both these functions are defined since we have supposed that the cycle of the orbit contains at least two symbols respectively associated to the two possible movements.

We have then the following theorem:

**Theorem 3.7** *Let us define the function $N$ from $O^*$ into $O^*$ by:*

$$\begin{aligned} N(\varepsilon) &= next(\flat) \\ N(o_i.s) &= \Sigma_{j=1}^{N(s)} next(succ^{j-1}(o_i))) - N(s) \end{aligned}$$

*We will have $N(o_i.s) = 0$ whenever $N(s) = 0$.*

*We define also the series $s_i$ and $a_i$ by induction by:*

$$\begin{aligned} s_0 &= \varepsilon \\ a_0 &= succ(\flat) \\ a_{i+1} &= succ^{N(s_i)}(a_i) \\ s_{i+1} &= a_i.s_i \end{aligned}$$

*Then in the $i$-th configuration of the form $(w_i.s, \flat, \varepsilon)$ attained by the machine $M$ in which $w_i$ exactly recovers the initial zone and $s \neq \varepsilon$ is such that $s = s_i$.*

**Proof** This theorem enables us to compute the series of the configuration of the divergence zone. We will prove this theorem by induction on $i$. In fact, we will prove also another one claim: $N(s_{i-1})$ is equal to the number of times the head of the machine leaves the divergence zone between the configurations $(w_{i-1}.s_{i-1}, \flat, \varepsilon)$ and $(w_i.s_i, \flat, \varepsilon)$.

We pose $(w_i.f_i, \flat, \varepsilon)$ the first configuration of the machine such that $l(f_i) = i$. Our purpose is then to prove that $f_i = s_i$ for all $i \geq 1$.

If $i = 1$, we have $s_1 = a_0.s_0 = succ(\flat)$. So $s_1$ is equal to the first symbol following $\flat$ in the orbit and such that its associated movement $R$. The configuration $(w_1.f_1, \flat, \varepsilon)$ is such that $f_1$ belongs to the orbit of $\flat$. So there exists $k \geq 1$ such that $c^k(\flat) = f_i$. The predecessor of $f_i$ in the orbit of $\flat$ has necessarily an associated movement equal to $R$ (since the machine has made a right movement after having written $f_i$ on the tape). This fact is expressed by the equation $m \circ c^{k-1}(\flat) = R$. At last it is easy that $k$ is the least number having such a property. Indeed let us denote $k_0$ the least number greater than 1 such that $m \circ c^{k_0-1}(\flat) = R$. If we suppose that $k > k_0$ then necessarily the machine admits a configuration of the form $(w, c^{k_0-1}(\flat), \varepsilon)$

where the string $w$ corresponds to the initial zone. At this point we obtain a contradiction since the next configuration is then $(w.c^{k_0}(\flat), \flat, \varepsilon)$. So we have $f_1 = succ(\flat) = s_1$.

Moreover $N(\varepsilon)$ is equal to the number of succesors of $\flat$ in the orbit which have a movement equals to $L$. So when leaving for the first time the initial zone, the head will reenter $N(\varepsilon)$ times the initial zone before to reach the second cell of the divergence zone that will make the configuration being $(w_1.s_1, \flat, \varepsilon)$. So both properties are established for the case $i = 1$.

For the general case, we will suppose that both properties are true for rank $i$ and try to establish it for the rank $i+1$. We are then interested by the rewritings between $(w_i.s_i, \flat, \varepsilon)$ and $(w_{i+1}.f_{i+1}, \flat, \varepsilon)$. We have $s_i = a_{i-1}.s_{i-1}$, so the first configuration is $(w_i.a_{i-1}.s_{i-1}, \flat, \varepsilon)$. If $N(s_{i-1}) = 0$, then the head will no more attain the cell corresponding to $a_{i-1}$. So we will have $a_i = a_{i-1} = succ^0(a_{i-1})$. Moreover if we pose $f_{i+1} = a_i.f$ then again the string $f$ only depends on the transition table of $M$ since it corresponds to a simulation of the machine with initial string $b.u.\flat$. So we have $f = s_i$ and the functionnal equality is respected. We have also $N(s_i) = 0$, which is conform with the fact that between $(w_i.s_i, \flat, \varepsilon)$ and $(w_{i+1}.s_{i+1}, \flat, \varepsilon)$, the machine does not quit the divergence zone.

Let us suppose now that $N(s_{i-1}) \neq 0$. Then from configuration $(w_i.a_{i-1}.s_{i-1}, \flat, \varepsilon)$ to configuration $(w_{i+1}.f_{i+1}, \flat, \varepsilon)$ the head of the machine attains $N(s_{i-1})$ times the first cell of the divergence zone, coming by the right. Each time such a thing happens the first cell of the divergence zone and the initial zone are modified until the head of the machine reenters again in the zone initially corresponding to string $s_{i-1}$. The machine enters this zone if and only if, at the previous rewriting the head is on the first cell of the divergence zone and writes a symbol whose associated movement is equal to $R$. That is to say that after the first reentry the first symbol of the divergence zone is $succ(a_{i_1})$, after the second one it is $succ^2(a_{i_1})$ etc ... Since the head reenters $N(s_{i-1})$ before to attain the configuration $(w_{i+1}.a_i.f, \flat, \varepsilon)$, we have necessarily with an obvious recurrence that $a_i = succ^{N(s_{i-1})}(a_{i-1})$. Moreover with the same argument as previously we have $f = s_i$. This establishes the first part of the theorem.

Let us count now the number of times the head quits the divergence zone from configuration $(w_i.a_{i-1}.s_{i-1}, \flat, \varepsilon)$ to configuration $(w_{i+1}.a_i.s_i, \flat, \varepsilon)$. At each way out of the zone initially corresponding to $s_{i-1}$, the head will not leave the divergence zone if the first symbol of the divergence zone has a $L$ associated movement. If the first symbol of the divergence zone has $q$ successors with a associated movement equal to $R$, then the machine will enter $q$ times the initial zone before that the head attains again the zone initially corresponding to $s_{i-1}$. The value of $q$ is given by function *next* decreased of 1. We know by hypothesis that the total number of reentries is equal to $N(s_{i-1})$. As a consequence, we have $N(s_i) = \Sigma_{j=1}^{N(s_{i-1})}(next(succ^{j-1}(a_{i-1})) - 1)$. This latter part establishes the whole theorem by recurrence. □

**Proof of theorem 3.6 (cont'd)** As we have seen the machine has five alternate possible behaviors: either it stops, or it loops, or it attains the dead zone and diverges by the left, or it attains an infinite number of in the divergence zone only or it attains an infinite number of configurations in both the initial and the divergence zone. The four first cases are semi-decidable and treated by the theorem 3.5. So the rest of the proof is devoted to establish the semi-decidability of the fifth case and, as a consequence, the decidability of the five cases.

The Turing machine will have the fifth behavior if and only if the head leaves by the right the initial zone an infinite number of times and $N(s_i) > 0$ for all integer value of $i$. As previously the first part of this criterion is decidable by Dirichlet's principle : since the initial zone have only a finite number of possible configurations, it is possible to make the complete graph of succession of configurations and then to decide whether it contains a cycle in which

there is at most one configuration with the head at the rightmost cell of the initial zone with an associated movement equal to $R$. After such a configuration, the machine enters the divergence zone. Then after a finite number of transitions the head enters the initial zone since we have supposed that $N(s_i) > 0$ for all $i$. As a consequence in the graph of succession of the configurations of the initial zone, the successor of the previous configuration will be the one corresponding to the transition function but with no movement at all (that is to say that the head stays on the last cell of the initial cell). At this point, in order to establish the semi-decidability of the fifth case it suffices to prove the semi-decidability of the property $N(s_i) > 0$ (indeed if two properties $P$ and $Q$ are semi-decidable then also is $P$ **and** $Q$).

We may remark that from the definitions, we have $\sum_{j=1}^{R(M)}(next(succ^{j-1}(o))-1) = L(M)$ for any symbol $o$ of the cycle of the orbit having a $R$-movement. Indeed $next(succ^{j-1}(o))-1$ counts the number of symbols with associated movement equal to $L$ between the $j-1$-th symbol following $o$ in the orbit and which has an associated movement equal to $R$ and the following one which has the same property. So since there is $R(M)$ symbols having this property in the orbit the sum of any $R(M)$ consecutive such expressions counts all the symbols with a $L$-movement, whence the result.

Now let use suppose that $a_i$ belongs to the cycle of the orbit. This property holds after a rank $i_0$ if we suppose that $N(s_i) > 0$ for all $i$. Moreover we will suppose that $L(M) > R(M)$ and $N(s_i) \geq L(M).R(M)$. We have then:

$$\begin{aligned} N(s_{i+1}) &= \sum_{j=1}^{N(s_i)} next(succ^{j-1}(a_i)) - N(s_i) \\ &= \sum_{j=1}^{N(s_i)} (next(succ^{j-1}(a_i)) - 1) \\ &= A_i.L(M) + \sum_{j=A_i.R(M)+1}^{N(s_i)} (next(succ^{j-1}(a_i)) - 1) \end{aligned}$$

if we pose $A_i = \lfloor \frac{N(s_i)}{R(M)} \rfloor$. But we have $A_i > \frac{N(s_i)}{R(M)} - 1$ and so:

$$\begin{aligned} A_i.L(M) &> \frac{N(s_i)}{R(M)}.L(M) - L(M) \\ &> \frac{N(s_i)}{R(M)}.(R(M)+1) - L(M) \\ &> N(s_i) + \frac{N(s_i)}{R(M)} - L(M) \\ &> N(s_i) \end{aligned}$$

since $N(s_i) \geq L(M).R(M)$. So we can conclude that $N(s_i + 1) > N(s_i)$.

So for the case where $L(M) > R(M)$ the property $(\forall i, N(s_i) > 0)$ is decidable. Indeed, either $N(s_i) > L(M).R(M)$ with $i \geq i_0$. In this case, the property $N(s_i) > 0$ holds for every greater value of $i$. Either $N(s_i)$ is bounded. In this case, since the values of $N(s_{i+1})$ and $a_{i+1}$ depend only on $N(s_i)$ and $a_i$, it is possible to build the complete graph of the succession of the couples $(N(s_i), a_i)$.

So the global procedure for this case is to build the graph for all the configurations $(N, a)$ where $N \leq L(M).R(M)$ and to check by enumeration if the set of the configurations $(N(s_i), a_i)$ where $i \geq i_0$ consist in a cycle of the graph. If this is not true that means that the condition $N(s_i) \geq L(M).R(M)$ is met from a certain rank.

Now, if we wish to majorate $N(s_i)$, we have:

$$\begin{aligned} N(s_{i+1}) &= A_i.L(M) + \sum_{j=A_i.R(M)+1}^{N(s_i)} (next(succ^{j-1}(a_i)) - 1) \\ &< A_i.L(M) + R(M).L(M) \end{aligned}$$

Indeed, the second term involves at most $R(M)$ summand each of them being strictly less than $L(M)$. Moreover we may note that $A_i \leq \frac{N(s_i)}{R(M)}$.

Now if we suppose $L(M) < R(M)$ and $N(s_i) \geq (R(M))^2.L(M)$, we have:

$$\begin{aligned} N(s_{i+1}) &< \frac{N(s_i)}{R(M)}.L(M) + R(M).L(M) \\ &< \frac{N(s_i)}{R(M)}.(R(M)-1) + R(M).L(M) \\ &< N(s_i) - \frac{N(s_i)}{R(M)} + R(M).L(M) \\ &< N(s_i) \end{aligned}$$

With this, it is also possible to decide. Indeed, here the set of values of $N(s_i)$ is bounded. So again by construction of a graph of transition it is possible to decide whether $N(s_i) > 0$ for all value of $i$.

The last case to treat is the one where $L(M) = R(M)$. This case is more delicate to handle and is dealed by congruences. Indeed in this case the more precise inequality that can be obtain is $N(s_i) - R(M) \leq N(s_{i-1}) \leq N(s_i) + R(M) - 2$ so that the global behavior of $N(s_i)$ cannot be decided this way. But here we have:

$$\begin{aligned} N(s_{i+1}) &= A_i.L(M) + \sum_{j=A_i.R(M)+1}^{N(s_i)} (next(succ^{j-1}(a_i)) - 1) \\ &= A_i.R(M) + \sum_{j=A_i.R(M)+1}^{N(s_i)} (next(succ^{j-1}(a_i)) - 1) \end{aligned}$$

since $L(M) = R(M)$. Moreover, for all integer $i$, we have $succ^{R(M)}(a_i) = a_i$. Indeed since there is $R(M)$ symbols in the cycle which have an associated movement equal to $R$, for every $o$ such symbol, we have $succ^{R(M)}(o) = o$. But every symbol $a_i$ is, by definition, the image of a symbol by the function $succ$ and so does have an associated movement equal to $R$. So we have:

$$\begin{aligned} N(s_{i+1}) &= A_i.L(M) + \sum_{j=A_i.R(M)+1}^{N(s_i)} (next(succ^{j-1}(a_i)) - 1) \\ &= A_i.R(M) + \sum_{j=1}^{N(s_i)-A_i.R(M)} (next(succ^{j+A_i.R(M)-1}(a_i)) - 1) \\ &= A_i.R(M) + \sum_{j=1}^{N(s_i)-A_i.R(M)} (next(succ^{j-1}(a_i)) - 1) \end{aligned}$$

and so we have:

$$N(s_{i+1})[mod\ R(M)] = \sum_{j=1}^{N(s_i)[mod\ R(M)]} (next(succ^{j-1}(a_i)) - 1)$$

Moreover, for all $i \geq 0$, we have $a_{i+1} = succ^{N(s_i)}(a_i) = succ^{N(s_i)[mod\ R(M)]}(a_i)$. So in this case the series $a_i$ and $N(s_i)[mod\ R(M)]$ may be also easily computed. Yet again we can build the complete graph of successions of the couples $(a_i, N(s_i)[mod\ R(M)])$. Since this graph has a finite number of vertices, it necessarily contains a cycle. If in this cycle no place is such that the term $N(s_i)[mod\ R(M)]$ is equal to 0 then we may conclude that $N(s_i) > 0$ for all i, since by definition we have $N(s_i) \geq 0$.

Let us suppose now that $N(s_i)[mod\ R(M)] = 0$, then $a_{i+1} = succ^{N(s_i)[mod\ R(M)]}(a_i) = a_i$ and $N(s_{i+1})[mod\ R(M)] = \sum_{j=1}^{N(s_i)[mod\ R(M)]}(next(succ^{j-1}(a_i)) - 1) = 0$. So by recurrence we have for all $j \geq i$, $a_j = a_i$ and $N(s_j)[mod\ R(M)] = 0$. But we also have $N(s_{i+1}) = A_i.R(M) = \lfloor \frac{N(s_i)}{R(M)} \rfloor .R(M)$. But since $N(s_i)[mod\ R(M)] = 0$, the division is exact and we have $N(s_{i+1}) = \frac{N(s_i)}{R(M)}.R(M) = N(s_i)$, and we can deduce that both series $a_j$ and $N(s_j)$ are stationnary in this case from a given time. So if $N(s_i) = 0$ then the head of the machine does not enter anymore the initial zone and if $N(s_i) \neq 0$, the head of the machine enters indefinitively the initial zone. That completes the theorem. □

# 4 Conclusion

In this article, we have made a study of the one-state Turing machines. We have seen that the halting problem is decidable for these machines while it is not for the two-states Turing machines (see [6]). In order to summarize we recall here the entire criterion: In order to make it clear we summarize the whole procedure:

- the machine may stop;
- the machine may loop;
- the head of the machine may enter the dead zone. Then the machine diverges by the left;
- the machine may attain two configurations $(w.s', \flat, \varepsilon)$ and $(w.b.s', \flat, \varepsilon)$ such that during all the rewritings between these two configurations the head enters the zone corresponding to the string $w$. Then the machine diverges by the right;
- If the orbit of $\flat$ contains a cycle the three following case are also possible:
  - If $L(M) > R(M)$ : After a finite number of transitions the first symbol of the divergence zone belongs to the cycle. Then you build the graph of successions for couples $(N, a)$, with $N$ integer such that $0 \leq N < L(M).R(M)$ and $a$ being any symbol belonging to the cycle of the orbit. If the set of the couples $(N(s_i), a_i)$ corresponding to the execution of the machine is totally included in the previous graph then you check whether any of these configurations is such that $N(s_i) = 0$. If at a given moment you have $N(s_i) \geq L(M).R(M)$ (i.e. the configuration does not belong any more to the graph), then the series $N(s_i)$ is stricly increasing and so non-null. If $N(s_i) = 0$ at a given moment then the machine diverges by the

right. If $N(s_i) \neq 0$ for all $i$ then we are in one of the first three cases except if set of the configurations of the initial zone is cyclic and contains at least one configuration from which the machine enters the divergence zone. In the latter case, the machine diverges by the right and enters indefinitively the initial zone and so does not stop.

- If $R(M) > L(M)$ : Here the number of values taken by $(N(s_i), a_i)$ is bounded. So in this case, we only have to simulate the machine during a finite number of transitions to decide whether the cycle of the possible configurations ends with a configuration such that $N(s_i) = 0$. At this point the final decision for the behavior of the machine is taken the same way as in the previous point.
- If $R(M) = L(M)$ : The study of the values taken by $(N(s_i)[mod\ R(M)], a_i)$ suffice to decide. Either the cycle of this graph does not contain any vertex such that $N(s_i)[mod\ R(M)] = 0$, and in this case $N(s_i) \neq 0$ for all $i$. Either the cycle is reduced to an unique vertex for which $N(s_i)[mod\ R(M)] = 0$. Then the series $(N(s_i), a_i)$ is stationnary and the non-nullity of $N(s_i)$ may be decided. Yet again the final decision for the behavior of the machine is taken the same way as in the two previous points.

So this article establishes the exact boundary of the decidability of one-state Turing machine with one dimensionnal tape.

This characterization also points out the future directions of possible researches in this area: which of these two properties have to be released in order to render the halting problem indecidable ? The work of Shannon [6] proves that if we allow the machine to have two states then the general problem is undecidable. But as the reader may easily figure if the alphabet is reduced to a single symbol then whatever the number of states is, the problem is decidable. Moreover in the work of Shannon, no indication is given of the minimal number of symbols necessary to render the problem indecidable.

The second axis is to release the one-dimensionnality of the tape. Indeed, there is also interest with Turing machines operating on tapes multidimensionnal or toric for instance. One can wonder whether these machines with only one internal state have a decidable computability or not. The author here makes the conjecture that for these two latters examples of tape the halting can also be decided.

# 5 Acknowledgements

The author wishes to thank Philippe Darondeau, research director at the IRISA for his helpful remarks on this article.

# References

[1] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society*, pages 230–265, 1936.

[2] L.M. Pavlotskaia. Solvability of the halting problem for certain classes of turing machines. *Notes of the Academy of Science USSR*, 13:537–541, November 1973.

[3] M. Margenstern. Nonerasing turing machines: a frontier between a decidable halting problem and universality. *Theoretical Computer Science*, 129:419–424, July 1994.

[4] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[5] J.B. Dennis P.J. Denning and J.E. Qualitz. *Machines, Languages and Computation.* Prentice-Hall, 1978.

[6] C.E. Shannon. *Automata studies*, chapter "A Universal Turing Machine with two Internal States". Annals of mathematics studies, 1956.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399